IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Bhaskar Ghosh

Confirmation No.: 7312

Examiner: Hwa, S.

Serial No.: 10/824,887

Group Art Unit No.: 2169

Filed on: April 13, 2004

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

This Appeal Brief is submitted in support of the Notice of Appeal filed on September 18, 2008. A Panel Decision from Pre-Appeal Brief Review was mailed October 10, 2008. The time period for filing this Appeal Brief runs until November 18, 2008.

I. REAL PARTY IN INTEREST

Oracle International Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-25 and 27-29 have been finally rejected and are the subjects of this appeal. Claim 26 has been cancelled.

IV. STATUS OF AMENDMENTS

The claims were not amended after the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent Claim 1 and also claims 15-25 and 27-29, which are summarized below. The claims summarized below are annotated to cross-reference features of those claims to specific examples of those features disclosed in the specification. However, the annotations are not intended to limit the scope of the recited features to those specific examples to which the annotations refer.

Claim 1 recites (with reference annotations in parenthesis) a method for processing a database statement within a database server, the method comprising the steps of: receiving at the database server the database statement (par. [0026]); determining that at least one operation required by the database statement can be parallelized (par. [0027]); within the database server, generating a set of information about how to execute the database statement; causing a plurality of slave processes to perform said at least one operation by sharing the set of information with each (par. [0028]) slave process of said plurality of slave processes, wherein the set of information shared with each slave process includes (a) information about a task to be performed by said slave process, and (b) information about one or more tasks, to be performed by processes other than the slave process (par. [0028]), to execute the database statement (par. [0029] and [0068]); and sending to each slave process of said plurality of slave processes data (par. [0038]) that indicates which part (par. [0029]) of the set of information shared with the slave process represents the part of the at least one operation that should be performed by the slave process.

Claims 15-25 and **27-29** recite computer readable storage media (FIG. 3, storage device 310) that carry instructions for causing processors (FIG. 3, processor 304) to perform the steps of the methods of Claims 1-14, respectively.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 15-25 and 27-29 stand rejected as allegedly non-statutory under 35 U.S.C. § 101.

2. Claims 1-7, 11, 14-21, 25, and 29 stand rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over Reiner, U.S. Patent No. 6,289,334 (hereinafter "Reiner") in view of Borden et al. (US Patent No. 5495606 (hereinafter "Borden").

3. Claims 8-10 and 22-24 stand rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over Reiner in view of Borden and further in view of Brid, U.S. Patent Application No. 2004/0268227 (hereinafter "Brid").

4. Claims 12-13 and 27-28 stand rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over Reiner in view of Borden and further in view of Hallmark, U.S. Patent Application No. 5,857,180 (hereinafter "Hallmark").

VII. ARGUMENTS

A. Claims 15-25 and 27-29 Are Statutory under 35 U.S.C. § 101

Within the arguments herein, Applicant will avoid merely repeating arguments already of record. However, a brief summary of how this dispute arose may be helpful.

The Final Office Action asserted that Claims 15-25 and 27-29 are drawn to a form of energy, but that energy is not one of the four categories of invention and therefore this claim(s) is/are not statutory. The Final Office Action made several other characterizations of energy, mostly directed to what energy is not.

Applicants acknowledge that, under current case law, claims cannot be directed to "energy" or "signals". However, Applicant has pointed out to the Examiner that the rejected claims are not, in fact, "drawn to a form of energy". Rather, the claims are directed to a computer-readable storage medium.

It is respectfully submitted that anyone skilled in the art (and most people who are not skilled in the art) would understand that a medium, that is readable by a computer, and that is used for storage **is** a "computer-readable storage medium".

Examples of excerpts in which storage media is described include:

"A storage device 410, such as a magnetic disk, optical disk, or magneto-optical disk, is provided and coupled to bus 402 for storing information and instructions."

"Non-volatile media includes, for example, optical, magnetic, or magneto-optical disks, such as storage device 410. Volatile media includes dynamic memory, such as main memory 406."

If the Examiner is going to persist in this rejection, the Examiner should explain why, for example, a magnetic disk is not a "computer-readable storage medium". Does it not store data? Is it not readable by a computer? Is it not a "medium"?

If the Examiner really means to say that "computer-readable storage medium" is not explicitly defined in the text, the Examiner is correct. However, no explicit definition is required when terms are used according to their ordinary meaning. For example, the Specification also does not explicitly define "computer", or "network", or any number of other well-understood words. Failure to have express definitions for well-known terms is not a valid basis for any rejection. "Computer-readable storage medium" means, quite simply, a medium that (a) is readable by a computer, and (b) is used for storage.

Given the normal meaning of the term "computer-readable storage medium", it is clear that patent-ineligible "transmission media" is **not** covered by the claim. "Transmission media" does not qualify as a "storage medium" for the simple reason that transmission media does not **store** information.

Further, regarding transmission media, it is true that the specification gives examples of transmission media. However, this is irrelevant because the claims have been amended to cover only "storage" media, thereby excluding transmission media. It is improper to reject a claim under 35 U.S.C. §101 for subject matter that the claim doesn't even cover.

In responding to the above, the Advisory Action stated "a computer readable storage medium includes punchcards (paper) or carrier wave (paragraph 0077) is not hardware or physical article, the 101 rejection is maintained". Applicant respectfully disagrees.

These remarks reflect a misreading of Applicant's specification, which **does not** define a computer-readable **storage** medium to include a carrier wave. Rather, the Specification clearly places "carrier wave" in the category of "transmission media", which the claims expressly **do not** cover (because they are limited to **storage** media).

For at least the above reasons, Claims 15-25 and 27-29 are statutory within 35 U.S.C. § 101.

B. Claims 1-7, 11, 14-21, 25, and 29 are patentable over Reiner or Borden.

Applicant does not merely repeat herein arguments already of record. However, to clarify why the current rejections should be withdrawn, the remark section of the Advisory Action is reproduced hereafter (with original grammar and punctuation):

> Applicant argued that, "Reiner discloses referencing pnode types having to executor functions which share the same pnode data structure. However, it is unclear which portion of Applicant's claims this section is relevant to. Acknowledgement and clarification is respectfully requested. Examiner respectfully disagrees. From instant specification (**1**) disclosed the sharing (**2**) process as follow, "the slaves are to perform their assigned tasks based on the shared execution plan, all the constructs used for the execution of a parallel statement" (paragraph 0030). "On each remote node, the original statement is delinearized, reconstructed, and built into the cursor by one slave, and shared by all others on the same node (paragraph 0034). "Interesting parallel execution statistics are all available in the row sources of the shared plan across all slaves and can be aggregated either live or after a query finishes" (paragraph 0035). "The QSC and all slaves share or use the same parallel single cursor" (paragraph

99037). "The database server provides an infrastructure to store, access, and aggregate statistic across all the shared slave cursors" (paragraph 0072).

In response to Applicant's arguments (**3**), Reiner discloses since host variables described in the bind descriptor are not modified by query execution, and since they are referenced identically in all parallel subqueries (**4**) of the same pcursor, the root cursor's bind descriptor can be shared (**5**) by parallel subqueries (column 60, lines 30-44). The subcursor pnode functionality could potentially be decomposed to more than one specialized pnode types, but need not be. It is unique among pnodes types described thus far in having two executor functions which share the same pnode data structure. The master executor is called by the subcursor pnode's parent (e.g. master). The master and parallel executors can coordinate their work by means of semaphores, with the master checking to see whether a next row is ready whenever one is requested by the subcursor pnode's parent (to avoid a busy wait it may actually be preferable for the parent of the subcursor node to wait on semaphore of all its children (e.g. slave) until one is ready. In this case, the role of the subcursor's master executor would be to perform whatever manipulation of buffer pointers and resetting of semaphores (e.g. restricting access to shared resources) is necessary to return a row to the parent, to keep the details of the subcursor's buffer and semaphore management transparent to the parent, and to factor out these functions from the different possible parent types. The master's role is somewhat analogous to that of a client-side DBMS software in a client-server DBMS. Conceptually, these tasks could be performed by the parent, so that the master executor is not strictly required.)(column 42, lines 19-46). (Advisory Action, page 2, numbering added)

In the Response filed August 14, 2008, Applicant requested clarification of which portions of Reiner correspond to the various steps and elements within Claim 1. The Advisory Action responded not by discussing Claim 1, but instead by quoting from Applicant's specification, as shown in (**1**) above.

However, these quotations are not responsive to Applicant's inquiries. The various rejections of the claims still do not make sense, are unclear, and are not correlated with the specific language of the claims. The above remarks within the Advisory Action do not change that fact.

Regarding (**2**) above, Applicant notes and agrees that Applicant's specification describes "sharing", and that Reiner also describes some type of "sharing". However, a dictionary also contains the word sharing. This will be discussed in more detail with (**5**) below. The main point is that despite repeated requests from Applicant, the various Office Actions still have not clarified which elements of Reiner or any other reference correspond with the various

portions of Applicant's claims. As an example, notice how the above quotation from the Advisory Action never once mentions any portion of Claim 1. Instead, the quotation discusses Applicant's specification, and then discusses various portions of Reiner, but without ever referencing any claim.

Regarding (**3**) above, the remarks about Reiner do not appear to be "in response to" any of Applicant's arguments. Applicant is forced to guess at the reasoning of the rejection.

Regarding (**4**) above, Reiner's host variables being "referenced identically in all parallel subqueries" is a good example of what Applicant is *not* claiming. Note specifically Claim 1 section (b), and the language "information about one or more tasks, to be performed by processes other than the slave process, to execute the database statement". This claimed feature is contradicted by Reiner's language "referenced identically".

Regarding (**5**) above, it is unclear whether Reiner's "root cursor's bind descriptor **can be shared** by parallel subqueries" is meant to correspond with the claimed "**sharing** the set of information with each slave process of said plurality of slave processes". Applicant notes that both phrases contain some form of the word "share", but the resemblance ends there. Originally, the rejection appeared to depend on the Borden reference for the claimed "sharing". However, Applicant made various arguments in this area. Applicant cannot determine whether the rejection is now based on the notion that Reiner suggests the claimed "sharing".

Despite the attempts at clarification within the Advisory Action, the rejection of Claim 1 continues to be unclear. Specifically, the Examiner has yet to identify exactly which portion of Reiner allegedly corresponds to the claimed "slave processes". The rejection of Claim 1 also continues to be unclear as to exactly which prior art, and which portion therein, allegedly corresponds to the claimed "set of information", and also the claimed "sharing". Either way, neither Reiner nor Borden has any equivalent for the claimed "sharing the set of information with each slave process of said plurality of slave processes".

For convenient reference, a portion of Claim 1 is repeated herein.

> . . . causing a plurality of slave processes to perform said at least one operation by
> **sharing** the set of information with each slave process of said plurality of slave processes, wherein the set of information shared with each slave process includes
> (**a**) information about a task to be performed by said slave process, and
> (**b**) information about one or more tasks, to be performed by processes **other than** the slave process, to execute the database statement; and
> sending to each slave process of said plurality of slave processes data that indicates which part of the set of information **shared** with the slave process represents the part of the at least one operation that should be performed by the slave process. (emphasis added)

As stated, the various rejections of Claim 1 are confusing in that it is unclear exactly which portion of Reiner is being equated to the claimed "slave processes". The various Office Actions do not specify which of Reiner's subqueries, threads, subcursor, executor functions, or pnodes could potentially be the claimed "slave processes". In addressing this portion of the claims, the Office Action notes that Reiner discloses referencing pnode types having two executor functions which share the same pnode data structure (Final Office Action, Page 5, Paragraph 3; referencing Reiner's col. 42, lines 16-25). However, it is unclear which portion of Applicant's claims this section is relevant to.

The Final Office Action cites Reiner for "having two executor functions which share the same pnode data structure" but does not state which portion of Claim 1 this corresponds to. Applicant claims, inter alia, ". . . sharing <a> set of information with **each** slave process of said plurality of slave processes". It is possible that the Office Action intends for Reiner's two executor functions to correspond to the claimed "slave processes", and for Reiner's pnode data structure to correspond with the claimed "set of information".

If so, there is a flaw with such correspondences. Applicant is not claiming merely sharing some information with all slaves. Instead, as shown within the (a) and (b) sections highlighted above, the information shared with one particular slave contains information about

that the various tasks being performed by **other** slaves. Reiner's executor functions do not have such an arrangement with Reiner's pnode data structures.

The Final Office Action describes how Reiner's host variables are referenced identically in all parallel subqueries of the same pcursor (Final Office Action, Page 5, bottom paragraph), but again, it is not clear how this has any bearing on Claim 1. Although unclear, it is possible that Reiner's parallel subqueries are meant to correspond with the claimed slave processes, and the host variables are meant to correspond with the claimed "information". Such an assumption is reinforced by a quote from Reiner on page 6 of the Office Action, which states that a root cursor's bind descriptor "can be shared" by parallel subqueries (quoting Reiner's column 60, lines 31-61). Thus, the assumption that Reiner's parallel subqueries correspond to the claimed slave processes is doubly reinforced, although still never specifically stated. Thus, again, Applicant is forced to construct the rejection themselves from the various vague and unconnected remarks within the various Office Actions.

Supposing the above assumptions about Reiner's parallel subqueries are true, using Applicant's best attempt at understanding the rejection, these parallel subqueries still do not match up with the claimed slave processes. The host variables are described as being "referenced identically in all subqueries", while the claimed information comprises (a) information about a task to be performed by that slave process, and (b) information about one or more tasks to be performed by processes **other than** that slave process. Thus, the claimed information referenced in (a) and (b) is never "referenced identically" among all slave processes. Again, as stated, Applicant is not claiming merely sharing some information with all slaves. Instead, as shown within the (a) and (b) sections of Claim 1, the information shared with one particular slave contains information not only about the task being performed by itself, but also information about that the various tasks being performed by other slaves.

The Final Office Action cites Borden for the portion of the claim which recites "sending to each slave process of said plurality of slave processes data that indicates which part of the set of information **shared** with the slave process represents the part of the at least one operation

that should be performed by the slave process" (emphasis added). Applicant understands and acknowledges that the Final Office Action is not using Borden to suggest the claimed "set of information". However, it is not permissible to overlook that even this last clause of Claim 1 also contains the word "shared" as highlighted above. This is significant because within Borden, information/queries are always split, but are never "shared". Borden's slave processors 20B-20H and 52-53 never share any information, and thus cannot suggest the claimed step.

From the above it is apparent that no matter what explanation is made, neither Reiner nor Borden are suitable for supporting a rejection of Claim 1. Applicant has shown how various portions of the cited prior art are not applicable to and sometimes contradicted by the language of Claim 1.

For at least the above reasons, Claim 1, as well as all claims dependent therefrom, is patentable over Reiner in view of Borden under 35 U.S.C. § 103(a).

By virtue of their dependence from Claim 1, Claims 2-25 and 27-29 inherit the features that are distinguished from Reiner and Borden. Consequently, Claims 2-25 and 27-29 are also patentable, taken either individually or in combination, under 35 U.S.C. § 103(a).

As a secondary position, Applicant requests that prosecution be re-opened and an Office Action mailed that identifies with specificity which portions of the prior art correspond with the various claimed elements.

C. Claims 8-10 and 22-24 are patentable over Reiner, Borden, or Brid

Claims 8-10 and 22-24 are believed to be allowable based on their incorporation of limitations from the Independent claim, as well as additional limitations that distinguish over cited art. Further, the dependent claims introduce additional features that render them patentable over the prior art.

D. Claims 12-13 and 27-28 are patentable over Reiner, Borden, or Hallmark

Claims 12-13 and 27-28 are believed to be allowable based on their incorporation of limitations from the Independent claim, as well as additional limitations that distinguish over cited art. Further, the dependent claims introduce additional features that render them patentable over the prior art.

CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejections of Claims 1-25 and 27-29 lack the requisite factual and legal bases. Appellants respectfully request that the Honorable Board reverse the rejections of Claims 1-25 and 27-29.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

/ChristopherMTanner#41518/
Chris Tanner

Date: November 18, 2008

2055 Gateway Place, Suite 550
San Jose, California 95110-1083
Tel: (408) 414-1238
Fax: (408) 414-1076

VIII. CLAIMS APPENDIX

1. A method for processing a database statement within a database server, the method comprising the steps of:

receiving at the database server the database statement;

determining that at least one operation required by the database statement can be parallelized;

within the database server, generating a set of information about how to execute the database statement;

causing a plurality of slave processes to perform said at least one operation by

sharing the set of information with each slave process of said plurality of slave processes, wherein the set of information shared with each slave process includes

(a) information about a task to be performed by said slave process, and

(b) information about one or more tasks, to be performed by processes other than the slave process, to execute the database statement; and

sending to each slave process of said plurality of slave processes data that indicates which part of the set of information shared with the slave process represents the part of the at least one operation that should be performed by the slave process.

2. The method of Claim 1 wherein:

the step of sharing the set of information includes sharing an execution plan for the database statement; and

sharing the execution plan with a particular slave process of the plurality of slave processes is performed by:

providing an original statement of the database statement to a node on which the particular slave process resides, wherein the original statement is the form of the database statement in which the database statement was received by the database server;

at said node, generating an equivalent execution plan based on the original statement; and

the particular slave process accessing the equivalent execution plan.

3. The method of Claim 2 wherein:

further comprising the step of providing to the node additional information that includes at least one of (a) values associated with session parameters of a database session in which the database statement was received, and (b) values associated with optimizer parameters that were used by an optimizer to generate a plan for the database statement in a node other than said node; and

the step of generating an equivalent execution plan is performed based, at least in part, on the additional information.

4. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the step of sending to each slave process of said plurality of slave processes data that indicates which part of the at least one operation should be performed by the slave

process includes sending to each slave process data that indicates a specific portion of the execution plan that is to be performed by the slave process.

5. The method of Claim 4 wherein:

the step of sending to each slave process data that indicates a specific portion of the execution plan that is to be performed by the slave process includes sending to a particular slave process data that indicates a particular portion of the execution plan that is to be performed by the particular process; and

the method further includes the step of the particular slave process determining how to execute the particular portion based, at least in part, on characteristics of the execution plan other than the particular portion of the plan that is to be executed by the particular slave process.

6. The method of Claim 1 wherein:

the step of generating a set of information includes

generating an execution plan for the database statement;

constructing a shared cursor for the database statement, wherein the shared cursor provides access to the execution plan; and

the step of sharing access includes providing each slave process of said plurality of slave processes access to the shared cursor.

7. The method of Claim 6 wherein the step of providing each slave process of said plurality of slave processes access to the shared cursor includes allowing two or more of said slave processes to access a shared instance of the shared cursor.

8. The method of Claim 6 wherein the step of providing each slave process of said plurality of slave processes access to the shared cursor includes allowing one of the slave processes to access a first instance of the shared cursor, and allowing another one of the slave processes to access a second instance of the shared cursor.

9. The method of Claim 8 wherein:

the one slave process resides on a first node;

the other slave process resides on a second node; and

the first node is a different node than said second node.

10. The method of Claim 9 wherein:

a first plurality of slave processes on said first node share access to said first instance of said shared cursor; and

a second plurality of slave processes on said second node share access to said second instance of said shared cursor.

11. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the method further comprises the step of inserting into the execution plan a granule iterator row source that encapsulates a horizontal partitioning of a base object upon which the database statement operates.

12. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the method further comprises the step of inserting into the execution plan at least one distribution row source that specifies how data is to be redistributed between one of

a first slave set and a query coordinator; and

a first slave set and a second slave set.

13. The method of Claim 12 wherein the step of inserting into the execution plan at least one distribution row source includes:

inserting into the execution plan at least one sender-side distribution row source that indicates how sending processes are to distribute data that the sending processes produce; and

inserting into the execution plan at least one receiver-side distribution row source that indicates how receiving processes are to obtain data that the receiving processes are to consume.

14. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the method further comprises the step of inserting into the execution plan a parallelizer row source that encapsulates the scheduling of tasks that slave processes are to perform.

15. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.

16. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 2.

17. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 3.

18. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 4.

19. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 5.

20. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.

21. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.

22. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.

23. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.

24. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 10.

25. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 11.

26. (cancelled)

27. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 12.

28. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 13.

29. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 14.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.